FINAL REPORT

Efficient Coding of Video Images

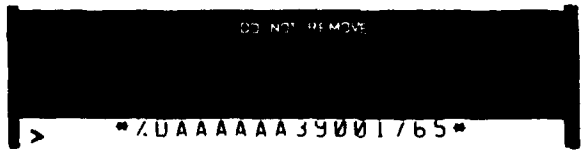
Department of the Army
Contract MDA 903-82-K-0521

covering the period
1 June 1982 - 31 January 1984

Submitted by
David H. Staelin

11 October 1984

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Research Laboratory of Electronics
Cambridge, Massachusetts 02139



SUMMARY OF RESEARCH

A. Video Test Facilities Development

In order to generate and display test sequences of videophone head-and-shoulders images, an input/output interface for an existing Data General Nova-4 computer was constructed. Video data is input through a monochrome camera followed by an 8-bit A/D converter. This data is sampled at the desired rate and is fed to one of two 64-kB dynamic RAM frame memories. The contents of these two memories are fed alternately to the computer DMA and thence to the 96 k-word Nova-4 memory preparatory to real-time storage on the 130 MB disk. The disk can record or playback at $\sim$ 320 kBps, which is the principal constraint on image size and frame rate.

To display the reconstructed video sequences, the data path is reversed. The display mode provides for optional real-time digital interpolation in both the horizontal and vertical directions so that small images (e.g. 64×60 or 128×120) can be substantially enlarged for display.

The system is capable of handling 8-bit monochrome images with 64 or 128 pixels per row and with 60, 120, or 240 rows per frame. The disk transfer rate supports the recording or display of 128×120-pixel images at 15 frames per second, or the equivalent. One mode employs 4 bits per pixel, which increases the available resolution or frame rate. For example, 256×240-pixel images can be displayed at 7.5 frames per second in this 4-bit mode.

Sequences of 128×120 pels are displayed with a 256×240 bilinearly-interpolated 8-bit format; this format removes the objectionable "blockiness" of the square raw pixels. The 256×240 4-bit images are bilinearly interpolated to 512×480 pels in real time. The camera mode permits recording of video data at the same real-time rates.

B. Morphological Coding of Images

The ambitious objectives of this program (i.e. the best feasible 9.6 kbps videophone images) has led us to pursue both aggressive and more traditional approaches to image coding. The first aggressive approach was morphological coding. This algorithm seeks to decompose each frame of the video image into "strokes" resembling the brush strokes of an artist. This is not a segmentation process, rather the strokes may overlap and are linearly superimposed to form the total image.

In this stroke-decomposition approach each scan line of each image is first efficiently filtered digitally so as to produce five different outputs which are then separately examined for local extrema larger than a variable threshold. Extrema clustered in local regions of the scan line are then examined and used to estimate the amplitude, shape, and position of each significant "stroke" crossing that scan line. These estimated parameters for each scan line are then assembled into two-dimensional strokes that efficiently characterize the desired image.

Figure 1 illustrates the present performance of these algorithms for a 128×120-pel image of a man's head and shoulders. Figure 1a is the original image, Figure 1b is an image wherein each line is independently decomposed into morphemes larger than some predetermined threshold, and Figure 1c is an image composed of overlapping two-dimensional luminance morphemes, a few of which are antisymmetric curvilinear "edge" functions, and the rest of which are symmetric curvilinear "gaussian" functions.

Many experiments have been run with these images employing different algorithms for decomposing and for representing the image. From these experiments it is increasingly clear that any such expansion of an image into quantized non-orthogonal functions introduces "ambiguity noise." We have identified

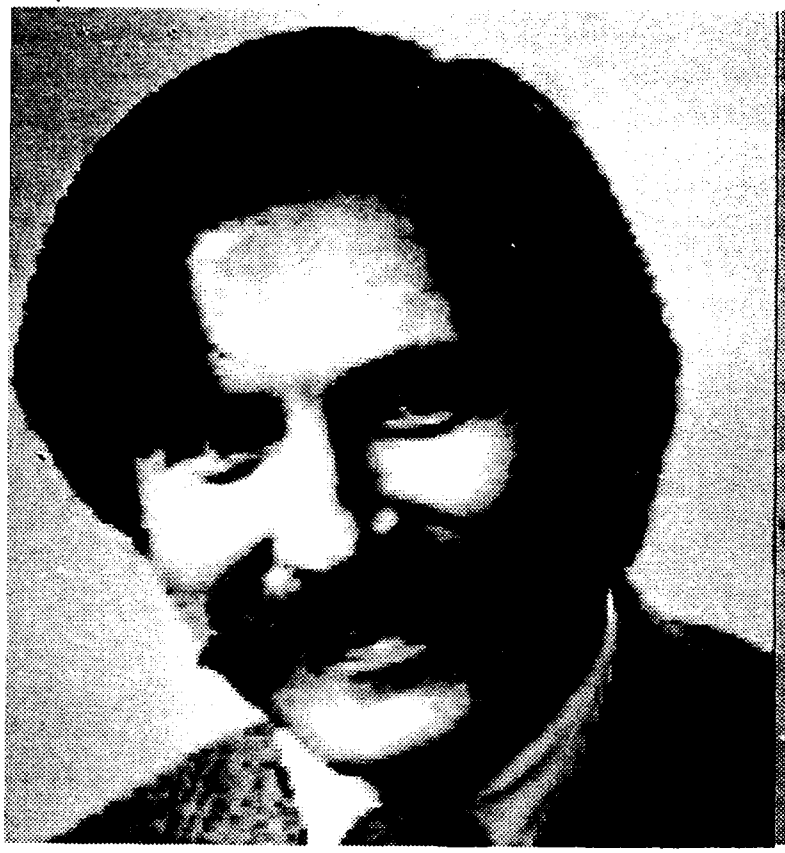


Figure 1(a)

Figure 1(b)

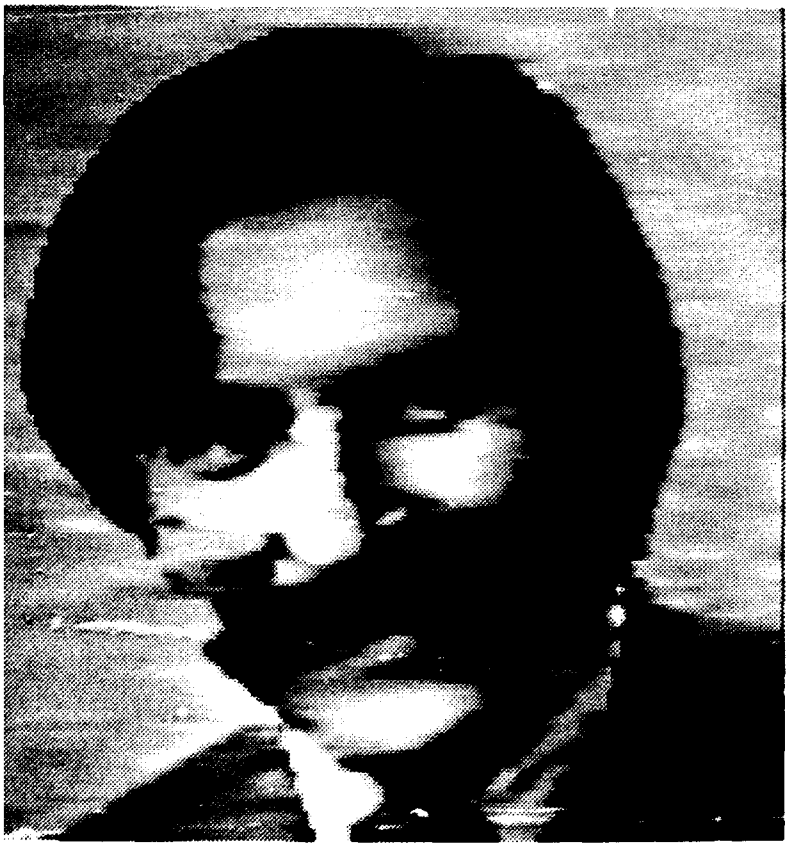


Figure 1(c)

Figure 1. (a) Original 120 × 120-pel 8-bit image displayed by an Autokon half-tone system. (b) Image where each line is independently represented only by gaussians and/or edge functions of standardized widths; horizontal streaks result from small variations in representation line-to-line (ambiguity noise). (c) Image composed only of gaussian and edge "strokes," which are curvilinear smoothly varying multi-line aggregates of related single-line functions. Sharp edge strokes bound the head and collar, whereas most other strokes are gaussians. Streaks and other ambiguity-noise generally arise at the isolated ends of strokes or at junctions between strokes. The most serious blemish here is the cheek discontinuity due to reduced local contrast between the face and background. Somewhat better image quality can be obtained here by lowering the error threshold and thus increasing the number of strokes.

two main types of ambiguity noise. The first is "detection ambiguity noise" which arises when a given intensity distribution can be represented equally well in two or more different ways and a choice must be made, each entailing a comparable but different error. Unless all the choices are made identically in any given region of an image, the spatial discontinuities in the error terms yield detection ambiguity noise. Efforts to homogenize decisions regionally still leave inter-regional discontinuities which are fortunately more tolerable.

"Representation ambiguity noise" arises when an image feature must be formed by superimposing two or more two-dimensional "stroke" morphemes; the noise arises at the junction of these strokes if the stroke ends don't blend perfectly. For example, consider the task of blending an antisymmetric edge stroke continuously into a symmetric gaussian, or consider the best representation for the junction of the letter "Y." Minimization of representation noise appears to be a non-trivial problem.

It appears that this image ambiguity noise is somewhat similar to the textual ambiguity noise that arose in early attempts to translate languages automatically. Both problems involve decisions to be made between ambiguous alternative representations of information, and ultimately synoptic criteria based on the total message, and perhaps involving some understanding of the message, may be necessary to yield acceptable system performance. Despite these early insights, however, it is still premature to conclude that stroke representations of images necessarily require excessive system "comprehension" in order to perform usefully. Attempts to understand and to reduce this ambiguity noise are under way. This work is continuing under a separate contract.

C. Two-Band Image Coding

This coding system is presently operational and continues to be improved. Its performance appears to be comparable to other selective replenishment coding schemes we have implemented, including adaptive DPCM and adaptive transform coding. For typical 128 × 120-pel head-and-shoulder images 1 bit per pel generally yields a good image if edges or other detail cover only about half the area. In general we find that fewer bits per pel suffice for larger images (e.g. 256 × 240) and somewhat more are required for smaller images.

In Figure 2 are presented two consecutive 128 × 102-pel images in a rapid-motion sequence; the two originals are in Figures 2a and b, and the coded version of 2b is shown in Figure 2c. The coded version required only 9760 bits to update its predecessor; no degradation is evident here. These images are simple Xerox reductions of computer line-printer graphics, and have much higher quality and recognizability when displayed on the video monitor.

The algorithm averaged the image in 6 × 6-pel blocks, and these resulting averages were bilinearly interpolated to produce the "low's," which is a blurred version of the original image; the difference between the original and the low's is the "high's." Superposition of the low's and high's would yield the original image exactly. The low's are quantized to 8 bits and the high's are quantized to 1 or 2 bits, with the quantization levels being chosen independently from one 6 × 6 block to the next. The 9760-bit data budget for the image of Figure 2c was: 1) 8232 bits (2 bits/pel) for high's, 2) 588 bits for low's for the same blocks, 3) 318 bits for low's for blocks where the high's are not to be updated, and 5) 622 bits for overhead such as the specification of quantization levels, block addresses, etc.

Figure 2(a)

Figure 2(b)

Figure 2(d)

Figure 2(c)

Figure 2. (a) Original 128 × 102-pel 8-bit image displayed (somewhat degraded) on an Anadex computer graphics printer. (b) The image following (a). (c) A two-band adaptively coded version of (b) using two-bit "high's" and selective replenishment; 9760 bits were employed without entropy coding. (d) A bilinearly interpolated 120 × 120-pel version of a 60 × 60-pel image (image (b)).

A video sequence with large interframe motion such as in Figure 2 would require approximately 8×9760 bits per second, or $\sim$ 78 kbps. These images are generally very good and fully recognizable when displayed on the video monitor. The data rate would drop to $\sim$ 50 kbps for 1-bit high's, or for more reasonable motion. Use of one-bit high's for a $\sim 60 \times 60$-pel image with modest motion may permit 9.6 kbps data rates if entropy coding ($\sim$ 30 percent savings) is used. In Figure 2d a 60×60-pel image is illustrated for reference.

Experiments were also performed to bound the potential savings available from Huffman coding of 1-bit and 2-bit high's. It was found that entropy coding conditional on previous pixels was best, particularly if the data blocks were square. The savings ranged from 25 to 40 percent for blocks of high's having 6 to 12 pels, depending on the shape of the block, the use of causal predictive pels, and the number of bits per pel.